AFOSR-TR-97-0606

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 96	Final Technical Report 15 Sep 92 to 14 May 96

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Connectionists Models for Intelligent Computation	F49620-92-J-0519
6. AUTHOR(S) H. H. Chen	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Laboratory for Plasma Research University of Maryland College Park, MD 20742	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR/NM 110 Duncan Avenue, Ste B115 Bolling AFB, DC 20332-8050	10. SPONSORING/MONITORING AGENCY REPORT NUMBER F49620-92-J-0519

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

19971204 162

13. ABSTRACT *(Maximum 200 words)*

We have successfully demonstrated the capability of neural net to learn the generation grammar and automata for discrete symbolic sequences. The first level of complexity is represented by the regular grammar and can be learned by a recurrent neural net. The capacity of these recurrent neural net to represent finite state machines that generate these regular grammar has also been theoretically estimated. In the next level of complexity, neural net was trained to operate a stack memory to recognize context free sequences. Finally, we showed that recurrent net can be constructed with a neural tape to represent a universal Turing machine.

For continuous time series, we also showed that neural net can be used to classify curves with different topology, to control chaos system without pre-knowledge of its fixed points and to perform system identifications for real physical systems. For the last task, we have successfully trained a neural net to simulate the flight dynamics of a helicopter UH-60. The system used is a MIMO model of recurrent net. And the helicopter model has six degrees of freedom, the vertical, side and forward speed, the pitch rate, roll rate, yaw rate and four control maneuvers, the lateral longitudinal, directional and the collective controls.

DTIC QUALITY INSPECTED 4

14. SUBJECT TERMS	15. NUMBER OF PAGES
	7
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. 239.18
Designed using Perform Pro, WHS/DIOR, Oct 94

FINAL REPORT

CONNECTIONIST MODELS FOR INTELLIGENT COMPUTATION

Grant#: F49620-92-J-0519

Submitted to

Dr. Marc Jacobs
Air Force Office of Scientific Research/
DARPA

by

H. H. Chen
Laboratory for Plasma Research
University of Maryland
College Park, Maryland 20742

Final Report

Final Report

Adaptive Neural Network Models for Intelligent Computations

Principal Investigators:

H.H. Chen

Department of Physics
University of Maryland
College Park, Maryland 20742

May, 1996

Abstract

We have successfully demonstrated the capability of neural net to learn the generation grammar and automata for discrete symbolic sequences. The first level of complexity is represented by the regular grammar and can be learned by a recurrent neural net. The capacity of these recurrent neural net to represent finite state machines that generate these regular grammar has also been theoretically estimated. In the next level of complexity, neural net was trained to operate a stack memory to recognize context free sequences. Finally, we showed that recurrent net can be constructed with a neural tape to represent a universal Turing machine.

For continuous time series, we also showed that neural net can be used to classify curves with different topology, to control chaos system without pre-knowledge of its fixed points and to perform suystem identifications for real physical systems. For the last task, we have successfully trained a neural net to simulate the flight dynamics of a helicopter UH-60. The system used is a MIMO model of recurrent net. And the helicopter model has six dynamical degrees of freedom, the vertical, side and forward speed, the pitch rate, roll rate, yaw rate and four control manuvers, the lateral, lontitudinal, directional and the collective controls.

I. Introduction:

In the past few years, we have demonstrated that recurrent neural network can be trained to work like a finite state machine that will generate symbolic sequence obeying certain regular grammars. With the help of a soft stack, the recurrent neural net can also be trained to work as a finite state controller that will use the stack to infer sequences that are generated from context free grammars.

Theses results prompted us to ask whether in continuous dynamics, such complex sequential behavior can also be captured by neural networks? In particular, can neural network identify the time series that are generated from given chaotic attractors, which in general exhibit very complex temporal behavior? One general characteristic of chaotic attractor dynamics is the dissipative nature of the system and the contraction of the phase space of the system asymptotically. This allows us to apply the synchronization as a measure to characterize the chaotic time series. We tested these ideas on several neural networks and chaotic dynamical systems and found preliminary results encouraging.

The general usefulness of neural networks in processing temporal signals could be tested on other real dynamical systems. This is demonstrated in our second topics studied during the past year. We secured real flight dynamics data for UH-60 helicopters. These data were used to perform system identifications of the helicopter system under four separate control mode. Our results showed that neural network modeling of the helicopter flight dynamics can be implemented easily without the complicated knowledge of the physical model of the helicopter. The performance of the neural network simulator is perhaps superior than the traditional system identification methods.

II. Chaotic Attractor Signal Classification using Neural Network Synchronizer

This work is stimulated by the interesting work of Pecora and Carroll about the synchronization of chaotic attractors. Our basic assumption is that a recurrent neural network could simulate any given dynamical systems. It should therefore be able to simulate chaotic dynamics.

1) Basic Recurrent Networks

We consider a recurrent network with N state-neurons $x(t)$. The dynamic equations can be written as a set of first order differential equations:

$$d/dt\, x(t) = F\,(x(t), w, I(t)) \qquad (1)$$

where w is a matrix representing the weights and other adjustable parameters, $I(t)$ is a vector representing the external inputs to the network at time t, and $x(t)$ is the state neuron vector. The nonlinear function $F()$ may be chosen as the sigmoid function.

There is an intrinsic difficulty in trying to use recurrent neural net to classify continuous temporal patterns, namely, there is no unique way to initialize the internal state neurons. This problem is absent in the grammatical inference problem because the sequential machine can be assumed to start from a common start state and end with a set of fixed end states. For real world signals, such as sonar signals, radar signals and other continuous signals, it is difficult to say when the signal started or ended. We therefore need a synchronization scheme to time align the signal with the internal states of the neural classifier. Since chaotic dynamical systems approach to a fixed attractor asymptotically, we would like to study the synchronization problem with them first.

2) Pecora and Carroll's work on Synchronization

Pecora and Carroll studied the synchronization of two nonlinear dynamical systems. One system is the driving system and the other one the driven system. For example, consider a three dimensional autonomous system, running under a certain initial condition,

$$d/dt\, x(t) = F(x, y, z), \qquad d/dt\, y(t) = G(x, y, z), \qquad d/dt\, z(t) = H(x, y, z)$$

has an orbit on the attractor given by $(x_1(t), y_1(t), z_1(t))$. A reduced system driven by the variable $x_1(t)$ of the given dynamical system is running under a different initial condition.

$$d/dt\, y(t) = G(x_1(t), y, z) \quad \text{and} \quad d/dt\, z(t) = H(x_1(t), y, z)$$

Would these two dynamical system eventually synchronize with each other? If it does, we can probably use it as a way to identify the signal $x_1(t)$ as coming from the given chaotic dynamical system without worrying about the initial assignment of internal state neuron activation values. Pecora and Carroll showed that under a very general conditions, namely that the Lyapunov exponent of the reduced dynamical systems are negative, the synchronization is possible. Furthermore, the two dynamical system does not have to be faithfully reductions from each other. As long as they are close in their parameters, the synchronization will be close.

3) Neural Network Simulations.

The synchronization idea can be considered as a generalization of the fixed point attractor idea of recurrent network dynamics first studied by Pineda. Now that the asymptotic attractors are no longer the simple fixed points which are trivially static. Our final attractors are full fledged chaotic attractors with complex time dynamics. The neural networks are demanded not only to reproduce these complex time dynamics but also to do it with locked time steps. In this sense, we are building a set of neural network templates for the various nonlinear dissipative dynamical systems. We would like to match a given unknown temporal signals with these neural network templates by measuring the degree of synchronization achieved to identify the signal.

To study the feasibility of this idea, we chose signals from the attractors of two different dynamical systems. The first one is the Lorenz attractor *(a=10, b=2.667, r=28)*

$$dx/dt = -a(x+y), \quad dy/dt = -xz + rx - y, \quad dz/dt = xy - bz$$

and the second system is the Rossler attractor: *(m=5.7, b=0.2)*

$$dx/dt = -(y+z), \qquad dy/dt = -x + by, \quad dz/dt = b + z(x - m)$$

The system parameters are chosen such that both of them have chaotic attractors.
The neural network chosen for these two attractors is a recurrent net with the following network dynamics:

$$x_i(t+1) = ax_i(t) + b\,[\,S\,W_{ij}y_j(t) + S\,W_{ijk}y_j(t)y_k(t) + q_i]$$
$$y_i(t) = \tanh\,(c_ix_i(t))$$

The input to the network is the one dimensional time series $x(t)$ sampled from the chaotic attractors to be classified which is clamped to the first component of the state vector $x_i(t)$ in the neural network. The synchronization error measure is defined as

$$E(\tau) = 1/\tau \int^{\tau}(RI(t) - x_1(t))^2\, dt$$

4) Results:

We used RTRL training algorithm to train the network with *5000* time steps data from each attractor. After about *10* epochs of training, the synchronization error reached a steady state value. We then tested the trained neural net synchronizers with new data. All the input data (length *5000* time steps) were correctly identified. These testing data are calculated from the original dynamical systems with different initial conditions. The synchronizers are also initialized with different internal state values. The results showed that the synchronization errors are not sensitive to these initial conditions.

These preliminary results showed that recurrent neural net can indeed be trained to simulate chaotic dynamical systems. Use them as synchronizer templates are robust for the limited number of experiments done so far. We intend to do more work along these lines and eventually go beyond the chaotic attractor dynamics to test on more general temporal signals.

III. Simulation of Helicopter Flight Dynamics

1) Introduction:

The main thrust of this research is to try constructing a neural net helicopter dynamics simulator based on flight data supplied to us by Captain Walker. Previously, a one variable recursive model was constructed to fit and predict the pitch rate q and its time derivative q dot from the flight database with fairly good result. All other state variables are included in this model and are imported directly from the recorded database. It is therefore an incomplete simulation. In the past four months, we attempted at complete simulations for four independent maneuvers separately. These simulations were done with full recursion of the participated state variables. The predictions of these complete recursive models are noticeably better than the previously constructed one-variable recursive models.

2) One Variable Recursive Models

These were our first attempt to test out the idea of a neural net helicopter flight dynamics simulator. Since the total number of the control and state dynamical variables are many(15 state variables and 4 control inputs), it seemed reasonable to test out the neural net simulator idea by letting only one state variable to evolve recursively with the predictor dynamics. All other control and state variables were imported directly from the recorded data set. This is therefore an incomplete simulation. The success of this first attempt prompted us to proceed with a more complete modeling.

3) Complete Recursive Models for Separate Control Maneuvers

In our database, there are four separate control maneuvers. These control maneuvers will have dynamical consequence on separate subsets of the 15 state variables. For example, the lateral control maneuver will mainly affect the rolling variable p, its time derivative pdot and the side shift velocity v and v dot. It is therefore interesting to see how good the neural net simulator will be by training and testing out on data belong to these separate set of maneuvers.

4) Variables Included

As we said earlier, in these control maneuver dependent simulations, only a subset of the *15* state variables are included in each simulation experiment. For example, in the simulation for which the maneuver is dominated by the lateral control input, the state variables included in the simulation input are :

State Variables: *p(t), p(t-1), p(t-2);* and the
Control Input Variables: *Lat(t), Lat(t-1), Lon(t), Lon(t-1), Dir(t), Dir(t-1), Col(t), Col(t-1).*

Here the state variables *p(t+1)* will be generated from the neural net predictor at the time step *t* and then feed into the net for the next time step *t+1* to recursively generate an output *p(t+2)*. In other words, the output of this neural net predictor is :

Increment of the State Variable: *delta p(t) = p(t+1) - p(t).*

We choose delta *p(t)* instead of *p(t+1)* to improve on the sensitivity of the change for effective training.

The control variables included are imported directly from the recorded database. Although all four control variables are included, but in the case of lateral dominated maneuver, the lateral control input is the most important.

5) Training and Testing Data Set:

We picked randomly one run of lateral frequency sweep, one run of doublet and one run of *2-3-1-1* for training and tested on other data set of all three types. The training is completely local, i.e. only input output pairs at a local time frame are used for training. Their relative ordering and distance in time are not known to the neural net. Long time or even short time correlations are not incorporated into the simulator. It is therefore very encouraging to see the good simulation result.

6) Testing Results:

We trained four independent neural net simulators for the four separate control dominated maneuvers. Their testing results on the testing data set are generally good. The predicted curve follows the recorded data curve closely. In most cases, the relative and the absolute error are small. This is remarkable since only a subset of state variables are included in the simulation and also the prediction is completely recursive. This indicates that the helicopter dynamics in these data are mostly separable into the four sets of control maneuvers. Within each of these maneuver, only a subset of state variables are important to determine the state of the helicopter. Perhaps in a more general flight conditions, such as transitions between different control modes, the nonlinear coupling between these subset of state variables become important. Otherwise, these separate submodules seem sufficient to describe most of the flight conditions.

Conclusion:

We have successfully demonstrated the capability of neural net to learn the generation of grammar and automata for discrete symbolic sequences. This include the regular grammar and the associated finite state machine, the context free grammar sequences and the associated pushdown automata. Finally, we have also showed that recurrent net can be constructed with a neural tape to represent a universal Turing machine.

For continuous time series, we demonstrated the ability for neural net to classify curves with different topology, to control chaotic systems without the pre-knowledge of their fixed points and to perform suystem identifications for real physical systems. For the last task, we have successfully trained a neural net to simulate the flight dynamics of a UH-60 helicopter.